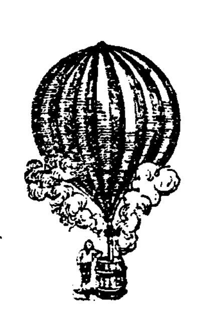


ARTINFO / MUSINFO

NOTE DE L'EDITEUR

Encore. Et toujours de la musique, avec ce numéro supervisé par Gilbert Dalmasso, où l'on verra apparaître un article de Daniel Fournier, ex-étudiant en informatique musicale. Un autre de LeClerc, un bien cher vieux complice, qui nous prouve combien la praxis des micros peut distiller encor des fragrances désuètes, aux confins méconnus de l'oulipoïdisme. Enfin l'éditeur lui-même se livre, pensif, à la retrospection sur un chapitre qui appartient désormais à l'avant-déménagement. Car Vincennes a migré au Nord. Réchauffons-nous un peu en citant deux vers du regretté Raoul Bertin, ce poète délicat (et tellement français) :



Si la réalité bien des fois nous ballonne,
Ne nous dégonflons point. Et poil à la banone.

J.A.

LE PROGRAMME VOLT

DANIEL FOURNIER

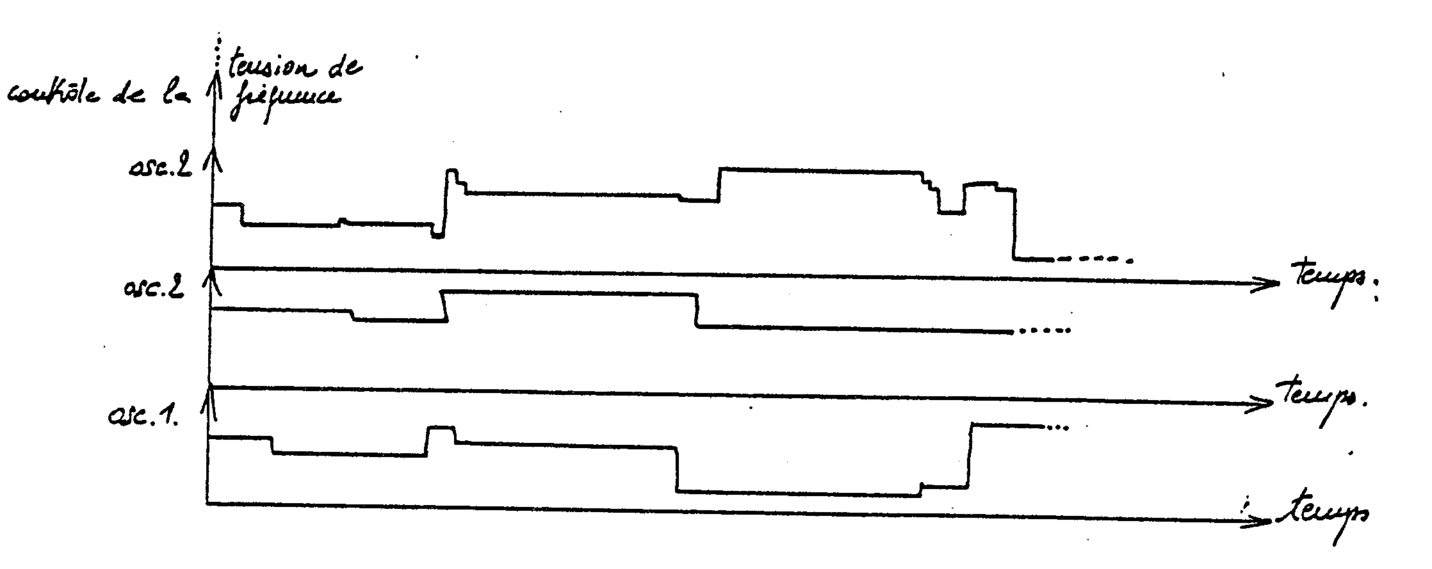
ARTINFO/MUSINFO N°30
1980

.Ce programme, rédigé en langage INTELGREU, est destiné au système de synthèse hybride du Groupe ART ET INFORMATIQUE de Vincennes (G.A.I.V.).

.Son but est d'obtenir des variations pseudo-aléatoires, indépendantes dans le temps, pour huit tensions de commande. Ces dernières sortant du 8 DAC viennent contrôler huit paramètres sur un ou plusieurs synthis VCS III.

<u>Exemple</u>   Huit oscillateurs (dont la fréquence est contrôlée par le 8 DAC).

Nous obtenons huit variations non-simultanées de la hauteur du son de chaque oscillateur:



.En contrôlant d'autres paramètres, tel que la durée de chute du son dans l'enveloppe ou la profondeur de la réverbération, nous pouvons donc modifier pseudo-aléatoirement l'évolution d'un ensemble de sons.

.Pour éviter une variation continue de tous les paramètres concernés, j'ai introduit une temporisation annexe qui permet de stopper pendant un court instant (maximum de 15 ms) l'évolution des variations. Par le boitier de clés de 8 bits, il est possible de faire varier cette courte halte suivant la valeur affichée sur les clés "04" - (00 pour le temps le plus long), (01 pour le plus court).

.Pour accorder les oscillateurs concernés par ces variations, j'ai introduit dans mon programme un filtre qui donne à toute sortie la valeur affichée sur les clés "05" lorsque celle-ci affiche 00. En effet, il s'agit d'un opérateur NØR et on sait que NDI 00 place la valeur 00 dans le registre A, valeur qui sort immédiatement sur toutes les sorties.

.Pour filtrer les tensions utilisées (ex.: voir programme DODECAPARAM), la même clé "05" sera utilisée, car suivant la valeur placée après une instruction NDI, on empêche certaines valeurs de sortir:

NDI FF : tout passe - instruction inopérante; clé " 05 " à FF

NDI F8 : ne passent que les valeurs se terminant par 0 ou 8; clé "05" à F8 (si l'on est accordé dans le tempérament: notes de la gamme)

NDI E0 : ne passent que les valeurs 00, 20, 40, 60, 80, A0, C0, et E0; clé "05" à E0.

etc...

PROGRAMME PRINCIPAL

(placé sur ruban)

TEMPORISATION ANNEXE

```
1Ø36 LHI 1Ø         Initialisation du registre H ——→ les valeurs seront effectivement
1Ø38 INI Ø4         (A)←—clé "04"                    rangées de l'adresse 10 D1 à
1Ø39 SUI Ø1         Décrémentation de (A)            l'adresse 10 E8 (à initialiser).
1Ø3B JFZ = 1Ø39     Test sur le contenu de A:
                             si (A) ≠ 00, bouclage en 1Ø39
                             si (A) = 00, on passe à l'instruction suivante
```

TESTS SUR L'ETAT DES COMPTEURS (temps de chaque paramètre)

```
      1Ø3E LLI D1
      1Ø4Ø LBM            exemple sur un compteur: celui rangé en 10D4
(A)   1Ø41 DCB                    qui correspond à la tension (B)
      1Ø42 CTZ = 1Ø81
      1Ø45 LMB
      ---------------
      1Ø46 LLI D4         appel de l'adresse 10D4
      1Ø48 LBM            chargement du contenu de la mémoire dans le registre B
(B)   1Ø49 DCB            décrémentation du contenu de B: (B)←—(B) - 1
      1Ø4A CTZ = 1Ø88     test sur le contenu de B: si (B) ≠ 00, continuer les instructions sui-
                                                    si (B) = 00, appel du sous programme vante
                                                          rangé en 1088
      104D LMB            chargement du nouveau
                          contenu de B (après décrémentation) dans la mémoire à la même adresse
                                                                                        10D4.
      ---------------
      1Ø4E LLI D7
      1Ø1Ø LBM
(C)   1Ø51 DCB
      1Ø52 CTZ = 1Ø8F
      1Ø55 LMB
      ---------------
      1Ø56 LLI DA
      1Ø58 LBM
(D)   1Ø59 DCB
      1Ø5A CTZ = 1Ø96
      1Ø5D LMB
      ---------------
      1Ø5E LLI DD
      1Ø6Ø LBM
(E)   1Ø61 DCB
      1Ø62 CTZ = 1Ø9D
      1Ø65 LMB
      ---------------
      1Ø66 LLI EØ
      1Ø68 LBM
(F)   1Ø69 DCB
      1Ø6A CTZ = 1ØA4
      1Ø6D LMB
      ---------------
      1Ø6E LLI E3
      1Ø7Ø LBM
(G)   1Ø71 DCB
      1Ø72 CTZ = 1ØAB
      1Ø75 LMB
      ---------------
      1Ø76 LLI E6
      1Ø78 LBM
(H)   1Ø79 DCB
      1Ø7A CTZ = 1ØB2
      1Ø7D LMB
      ---------------
      1Ø7E JMP = 1Ø38
```

Le travail est le même pour les huit compteurs rangés en 10D1, D4, D7, DA, DD, E0, E3 et E6

Dès que le compteur arrive à 00, le sous-programme correspondant se charge:

- .d'appeler un sous-programme de calcul
- .de sélectionner la voie sur laquelle doit sortir la nouvelle tension
- .d'envoyer cette nouvelle tension
- .de revenir dans le programme principal

A la fin de cette succession de tests, on trouve une instruct de saut inconditionnel qui renvoie en début de programme; le seul moyen de l'arrêter est donc d'enfoncer la touche "RESET" sur le pupitre de l'INTEL 8008.

SOUS-PROGRAMMES

```
    1Ø81 CAL = 1ØB9
A)  1Ø84 LAI 8Ø
    1Ø86 OUT Ø9
    1Ø87 RET

    1Ø88 CAL = 1ØB9
B)  1Ø8B LAI 4Ø
    1Ø8D OUT Ø9
    1Ø8E RET

    1Ø8F CAL = 1ØB9
C)  1Ø92 LAI 2Ø
    1Ø94 OUT Ø9
    1Ø95 RET

    1Ø96 CAL = 1ØB9
D)  1Ø99 LAI 1Ø
    1Ø9B OUT Ø9
    1Ø9C RET

    1Ø9D CAL = 1ØB9
E)  1ØAØ LAI Ø8
    1ØA2 OUT Ø9
    1ØA3 RET

    1ØA4 CAL = 1ØB9
F)  1ØA7 LAI Ø4
    1ØA9 OUT Ø9
    1ØAA RET

    1ØAB CAL = 1ØB9
G)  1ØAE LAI Ø2
    1ØBØ OUT Ø9
    1ØB1 RET

    1ØB2 CAL = 1ØB9
H)  1ØB5 LAI Ø1
    1ØB7 OUT Ø9
    1ØB8 RET
```

.*8 sous-programmes (pour chaque paramètre: 1 sous-programme)*

exemple: celui rangé en 1Ø8F; (tension(C)).

appel du sous-programme de calcul rangé en 1ØB9
sélection de la voie de sortie sur le 8DAC: ici: 2Ø
sortie de la valeur contenue dans A sur cette voie
retour au programme principal.

.les 8 sous-programmes sont identiques quant à leur fonction, excepté celle de sélectionner les 8 voies (de 80 à 01 pour les tension (A) à (H)).

.la tension qui sortira de chaque DAC est calculée en fonction de la précédente tension et d'un opérateur qui est lui-même différent à chaque nouvel appel du sous-programme de calcul pour éviter que la tension de sortie soit cyclique.

On aura donc:

une tension convertie donc une valeur dans A
un opérateur, une opération XØR

pour le <u>sous-programme de calcul de valeurs pseudo-aléatoires</u>:

-l'ancienne valeur de A est mémorisée
-l'ancienne valeur de l'opérateur de même.

	exemple: tension(A)		
1ØB9 INL	10D2		1) <u>calcul du nouvel opérateur et son rangement en mémoire</u>
1ØBA LAM	10D2	(A) ⟵ (M)	
1ØBB ADA	10D2	(A) ⟵ (A)+(A)	
1ØBC JFS = 1ØC1	10D2	saut si flag signe=0	
1ØBF XRI 7D	10D2	calcul (A)XØR 7D	
1ØC1 LMA	10D2	(M) ⟵ (A)	
1ØC2 INL	10D3		2) <u>calcul de la nouvelle valeur de A et calcul de la nouvelle valeur du compteur servant à la temporisation de chaque valeur de sortie et rangement en mémoire de la valeur de A</u>
1ØC3 XRM	10D3	calcul(A)XØR (M)	
1ØC4 RLC	10D3	rotation	
1ØC5 LBA	10D3	rangement (B) ⟵ (A)	
1ØC6 RLC	10D3	rotation	
1ØC7 RLC	10D3	rotation	
1ØC8 LMA	10D3	rangement (M) ⟵ (A)	
1ØC9 DCL	10D2		
1ØCA DCL	10D1		
1ØCB LMB	10D1	rangement (M) ⟵ (B)	
1ØCC LBA	10D1	rangement (B) ⟵ (A)	
1ØCD INI Ø5		(A) ⟵ clé "05"	
1ØCE NDB		filtrage (A)NØR (B)	
1ØCF OUT ØB		sortie sur 8 DAC	
1ØDØ RET		retour au sous-programme	

Dans l'exemple:
10D3: nouvelle valeur de A qui va sortir
10D2: nouvel opérateur qui a servi au calcul
10D1: nouvelle valeur du compteur.

Autre application: DODECAVOLT

.A l'aide du 4 DAC dont les gains sont préréglés pour une utilisation tempérée, et en modifiant certaines instructions dans les sous-programmes, on peut obtenir 4 tensions de commande en fréquence des 4 oscillateurs (filtre y compris) variant entre DO3 et SOL # 5.

.Si on prend les tensions (A), (C), (E), (G), il faudra:

d'une part, modifier le contenu du programme:

- en 10 CF: RET au lieu de OUT OB

- en 1084:	- en 1092:	- en 10A0:	- en 10AE:
OUT OC RET	OUT OD RET	OUT OE RET	OUT OF RET
pour (A)	pour (C)	pour (E)	pour (G)

d'autre part, filtrer la sortie de la valeur contenue dans A par la clé "05" positionnée en F8 (11111000), ce qui correspond à un calcul du style NDI F8 (laissant passer les valeurs terminant par 0 ou 8).

On pourra évidemment filtrer la sortie autrement pour jouer sur les demi-tons ou les quarts de tons ou encore pour jouer sur la gamme par tons.

ooooooo

Le Programme Param

Daniel Fournier

Artinfo/Musinfo n°30
1980

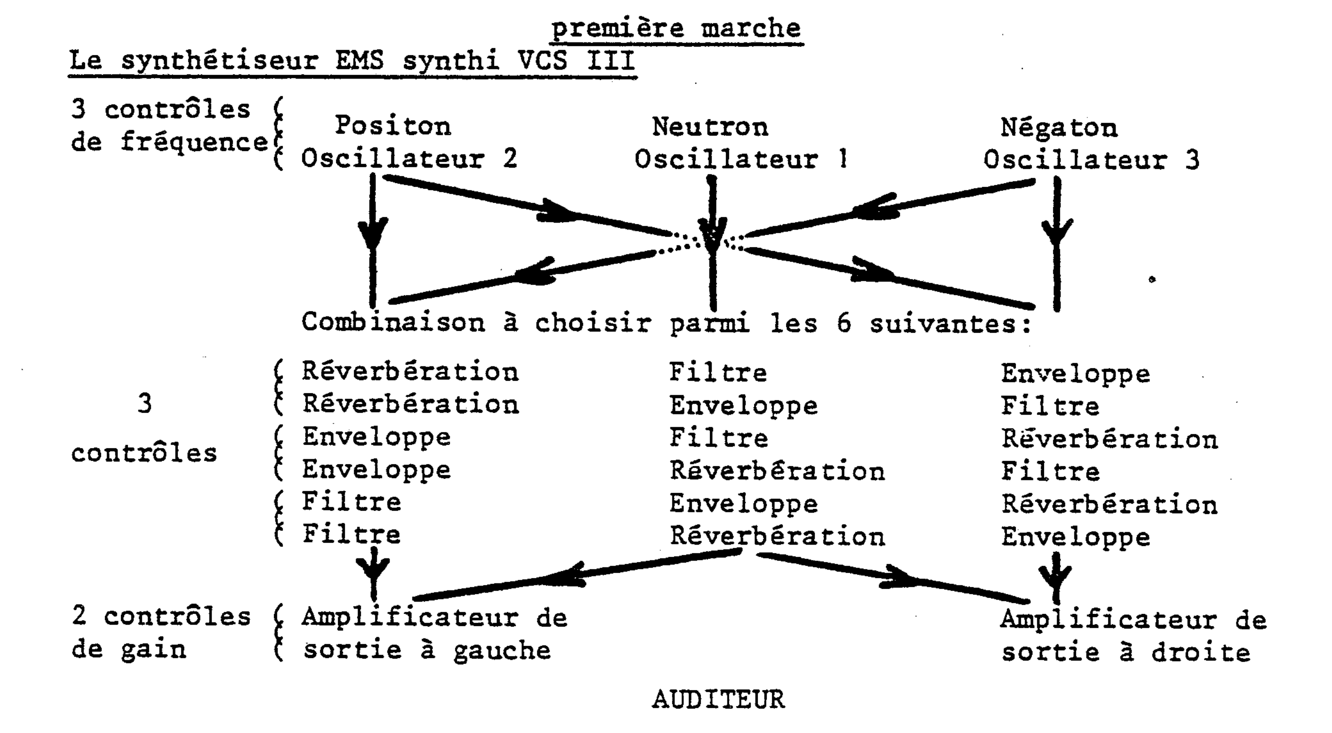
DEMARCHE
Persuader l'auditeur qu'il rêve à trois inconnus notoires jouant autour de lui leur chef d'oeuvre sans queue ni tête.

Positon, Négaton, Neutron, nantis de cornemuses dernier cri, s'obstinent, à toutes jambes, à déchiffrer leur partition respective selon leur fantaisie. Circulant au centre, Neutron "va, vient, fait l'empressé", départageant Positon et Négaton qui, comme le dit la légende, possèdent des atomes crochus..

DES MARCHES: pour résumer l'escalier.

première marche

Le synthétiseur EMS synthi VCS III



AUDITEUR

deuxième marche

Le 8-DAC: Convertisseur-digital analogique recevant des valeurs hexadécimales à la sortie de l'INTELLEC 8008, les convertissant en tensions électriques, lesquelles contrôlent les 8 paramètres sur le synthétiseur.

troisième marche

L'INTELLEC 8008: programmé en langage INTELGREU (merci Monsieur XRI 7 D) délivre des valeurs hexadécimales calculées de manière indépendante les unes des autres et recrée ainsi un caractère pseudo-aléatoire justifiant le chef-d'oeuvre sans queue ni tête, la fantaisie des inconnus et leurs déplacements désordonnés dans l'espace sonore.

DEFINITIONS

Nous avons huit paramètres ("A" à "H") à contrôler:

.variations de la fréquence des 3 oscillateurs
.variations de la fréquence de coupure du filtre
.variations du temps de la chute de l'enveloppe
.variations de la profondeur de la réverbération
.varations du gain des 2 amplificateurs de sortie,

donc huit valeurs de sortie dirigées vers le 8-DAC.

Nous avons quatre valeurs associées à chacun de ces paramètres:

	ex.paramètre"A"
1. Ø : valeur de l'opérateur de calcul	$Ø_A$.
2. K : valeur du compteur déterminant la durée relative de chaque sortie	K_A
3. k : valeur du compteur de temporisation qui sera progressivement décrémenté (k = K au moment du calcul)	k_A
4. S : valeur de la sortie	S_A

Le calcul donnant des nouvelles valeurs à partir des anciennes:

$$[2(Ø_{(n-1)})] \text{ XØR } (7D) = Ø_{(n)}$$
$$[RLC(K_{(n-1)})] \text{ XØR } (Ø_{(n)}) = K_{(n)} = k_{(n)}$$
$$[RRC(S_{(n-1)})] \text{ XØR } (Ø_{(n)}) = S_{(n)}$$

dès que $k_{(n)}$ atteindra 00, $Ø_{(n)}$, $K_{(n)}$ et $S_{(n)}$ seront considérés comme $Ø_{(n-1)}$, $K_{(n-1)}$ et $S_{(n-1)}$, le calcul reprendra.

remarque: ne pas omettre d'intialiser toutes ces valeurs lors du lancement ~~de la fusée~~ du programme.

DES FINITIONS:

Au moment de passer la seconde couche, il m'a semblé intéressant de placer un filtre NDI en sortie, accessible en temps réel par le jeu de clés "05", et ceci pour deux raisons:

../3

.régler le seul des paramètres:

.NDI 00 affecte la vleur 00 à chaque sortie, donc

-fréquences les plus grandes délivrées par les oscillateurs (les notes les plus aigues),
-fréquence maximale de coupure du filtre,
-profondeur de réverbération la plus grande,
-temps de chute de l'enveloppe le plus court,
-gain des amplificateurs de sortie le plus faible.

.filtrer éventuellement les sorties pour obtenir soit:

. sorties de notes précises (par ex. NDI FO gamme par ton si le 8-DAC est accordé suivant le tempérament)

.timbres précis (suivant certaines valeurs délivrées au paramètre qui contrôle la fréquence de coupure du filtre)

.profondeur de réverbération, gain de sortie, etc...crantés avec précision.

.ne pas oublier: NDI FF: instruction inopérante.

Au moment de passer la troisième couche, il m'a semblé préférable de pouvoir agir sur la temporisation générale; j'ai achevé ce programme en le complétant d'un sous-programme de temporisation annexe accessible en temps réel par le jeu de clés "04". La temporisation la plus courte (clé "01") va progressivement (ex.: "02",.."AA",..."FF) jusqu'à la plus longue (clé "00").

PROGRAMME PRINCIPAL

```
1036  LHI 10
----------------
1038  LLI DF        )
103A  LBM           }
103B  DCB           }  "A"
103C  CTZ = 108B    }
103F  CAL = 10FF    )

1042  LLI E3        )
1044  LBM           }
1045  DCB           }  "B"
1046  CTZ =1092     }
1049  CAL =10FF     )

104C  LLI E7        )
104E  LBM           }  "C"
104F  DCB           }
1050  CTZ = 1099    }
1053  CAL = 10FF    )

1056  LLI EB        )
1058  LBM           }
1059  DCB           }  "D"
105A  CTZ = 10A0    }
105D  CAL = 10FF    )

1060  LLI EF        )
1062  LBM           }
1063  DCB           }  "E"
1064  CTZ = 10A7    }
1067  CAL = 10FF    )

106A  LLI F3        )
106C  LBM           }
106D  DCB           }  "F"
106E  CTZ = 10AE    }
1071  CAL = 10FF    )

1074  LLI F7        )
1076  LBM           }
1077  DCB           }  "G"
1078  CTZ = 10B5    }
107B  CAL = 10FF    )

107E  LLI FB        )
1080  LBM           }
1081  DCB           }  "H"
1082  CTZ = 10BC    }
1085  CAL = 10FF    )

1088  JMP = 1038    )
```

8 SOUS-PROGRAMMES DE SORTIE

```
108B  CAL = 10C3 )
108E  LAI 80     }  "A"
1090  ØUT 09     }
1091  RET        )

1092  CAL = 10C3 )
1095  LAI 40     }  "B"
1097  ØUT 09     }
1098  RET        )

1099  CAL = 10C3 )
109C  LAI 20     }  "C"
109E  ØUT 09     }
109F  RET        )

10A0  CAL = 10C3 )
10A3  LAI 10     }  "D"
10A5  ØUT 09     }
10A6  RET        )

10A7  CAL = 10C3 )
10AA  LAI 08     }  "E"
10AC  ØUT 09     }
10AD  RET        )

10AE  CAL = 10C3 )
10B1  LAI 04     }  "F"
10B3  ØUT 09     }
10B4  RET        )

10B5  CAL = 10C3 )
10B8  LAI 02     }  "G"
10BA  ØUT 09     }
10BB  RET        )

10BC  CAL = 10C3 )
10BF  LAI 01     }  "H"
10C1  ØUT 09     }
10C2  RET        )
```

1 SOUS-PROGRAMME DE CALCUL

```
10C3  INL          }
10C4  LAM          }
10C5  ADA          }
10C6  JFS = 10CB   }  "Ø"
10C9  XRI 7D       }
10CB  LMA          }
10CC  LBA          }

10CD  INL          }
10CE  LAM          }
10CF  RLC          }  "K"
10D0  XRB          }
10D1  LMA          }

10D2  INL          }
10D3  LAM          }
10D4  RRC          }  "S"
10D5  XRB          }
10D6  LMA          }

10D7  INI 05       }
10D8  NDM          }  "Filtre"
10D9  ØUT 0B       }

10DA  DCL
10DB  LBM
10DC  DCL
10DD  DCL
10DE  RET
```

RANGEMENT 8 GROUPES DE 4 PARAMETRES EN MEMOIRE

```
10DF  (k)  }
10E0  (Ø)  }  "A"
10E1  (K)  }
10E2  (S)  }

10E3  (k)  }
10E4  (Ø)  }  "B"
10E5  (K)  }
10E6  (S)  }

10E7  (k)  }
10E8  (Ø)  }  "C"
10E9  (K)  }
10EA  (S)  }
```

rangement 8 groupes de 4 paramètre
en mémoire (suite)

```
10EB  (k)  }
10EC  (Ø)  }  "D"
10ED  (K)  }
10EE  (S)  }

10EF  (k)  }
10F0  (Ø)  }  "E"
10F1  (K)  }
10F2  (S)  }

10F3  (k)  }
10F4  (Ø)  }  "F"
10F5  (K)  }
10F6  (S)  }

10F7  (k)  }
10F8  (Ø)  }  "G"
10F9  (K)  }
10FA  (S)  }

10FB  (k)  }
10FC  (Ø)  }  "H"
10FD  (K)  }
10FE  (S)  }
```

1 SOUS-PROGRAMME DE TEMPO ANNEXE

```
10FF  LMB
1100  INI 04

1101  SUI 01
1103  JFZ = 1101
1106  RET
```

COMMENTAIRES RELATIFS AU DEROULEMENT DU PROGRAMME

avec, pour exemple, les parties de programme relatives au paramètre "A", à ne pas confondre avec (A) qui signifie contenu du registre A.

I. PROGRAMME PRINCIPAL

1036 LHI 10	: chargement immédiat de la valeur 10 dans le registre H, car les valeurs associées à chacun des paramètres ("A" à "H") sont rangées en mémoire de 10DF à 10FE.
1038 LLI DF	: chargement immédiat de la valeur DF dans le registre L, ce qui autorise l'appel de l'adresse 10DF de la mémoire contenant la valeur courante k_A (compteur de temporisation du paramètre "A").
103A LBM	: chargement de la valeur courante k_A dans le registre B.
103B DCB	: décrémentation du contenu du registre B.
103C CTZ = 108B	: appel conditionnel du sous-programme de sortie du paramètre "A" (rangé en 108B) selon l'état du contenu du registre B: si (B) ≠ 00, alors passer à l'instruction suivante (l'état des valeurs associées au paramètre "A" restent inchangées, en particulier la valeur de sa sortie S_A). si (B) = 00, alors appel du sous-programme de sortie; (B) = k_A=00 signifie que la valeur du compteur de temporisation est entièrement décrémenté, que les valeurs associées au paramètre "A" vont être recalculées pour une nouvelle sortie.
103F CAL = 10FF	: appel inconditionnel du sous-programme de temporisation annexe (rangé en 10FF) dans lequel aura lieu le chargement du contenu du registre B (qui vient d'être décrémenté) en mémoire: c'est le nouveau k_A.
1042 à 1085	: ensemble d'instructions concernant les sept autres paramètres ("B" à "H").
1088 JMP = 1038	: branchement inconditionnel en début de programme principal (ce qui revient à dire qu'il faut enfoncer la touche "HLT" sur l'INTELLEC 8008 pour en sortir ...).

II. SOUS-PROGRAMMES DE SORTIE

108B CAL = 10C3	: appel inconditionnel du sous-programme de calcul (rangé en 10C3) pour une nouvelle valeur de la sortie S_A.
108E LAI 80	: sélection de la voie du 8-DAC relative au paramètre "A".
1090 ØUT 09	: envoi de l'information S_A (contenue dans la voie 0B) dans la voie précédemment sélectionnée.
1091 RET	: retour au programme principal.
1092 à 10C2	: ensemble d'instructions concernant les sorties des sept autres paramètres ("B" à "H").

III. SOUS-PROGRAMME DE CALCUL

1 - CALCUL DE LA NOUVELLE VALEUR DE L'OPERATEUR DE CALCUL $Ø_{A(n)}$:

$Ø_{A(n)} \longleftarrow [2(Ø_{A(n-1)})]$ XØR 7D

10C3 INL : incrémentation du contenu du registre L permettant de se placer à l'adresse 10E0, où la mémoire contient la valeur $Ø_{A(n)}$ maintenant considérée comme $Ø_{A(n-1)}$, la valeur du compteur de temporisation k_A ayant atteint 00.

10C4 LAM : $(A) \longleftarrow Ø_{A(n-1)}$; chargement du contenu de la mémoire à l'adresse 10E0 dans le registre A.

10C5 ADA : $(A) \longleftarrow (A)+(A)$; addition du contenu du registre A sur lui-même.

10C6 JFS = 10CB : branchement conditionnel à l'adresse 10CB pour éviter l'opération "XØR" (OU exclusif) qui suit.
Saut si le flag de signe (bit de gauche du registre A modifié précédemment) est égal à 0, sinon passage à l'instruction suivante.

10C9 XRI 7D : $(A) \longleftarrow (A)$XØR 7D; opération logique "OU exclusif" entre le contenu du registre A et la valeur 7D, le résultat de cette opération donne $Ø_{A(n)}$.

10CB LMA : chargement de $Ø_{A(n)}$, contenu dans le registre A, en mémoire à la même adresse 10E0, de manière à la retrouver au moment du calcul de sa prochaine valeur.

10CC LBA : chargement de $Ø_{A(n)}$, contenu dans le registre A, dans le registre B pour remplir sa fonction d'opérateur dans les instructions suivantes.

2 - CALCUL DE LA NOUVELLE VALEUR DU COMPTEUR $K_{A(n)}$:

$K_{A(n)} \longleftarrow [RLC(K_{A(n-1)})]$ XØR $(Ø_{A(n)})$

10CD INL : incrémentation du contenu du registre L permettant de se placer à l'adresse 10E1, où la mémoire contient la valeur $K_{A(n)}$ maintenant considérée comme $K_{A(n-1)}$, la valeur du compteur de temporisation k_A ayant atteint 00.

10CE LAM : $(A) \longleftarrow K_{A(n-1)}$; chargement du contenu de la mémoire à l'adresse 10E1 dans le registre A.

10CF RLC : rotation à gauche du contenu du registre A (chaque bit du registre A se déplace vers la gauche, le plus à gauche se retrouve le plus à droite).

10D0 XRB : $(A) \longleftarrow (A)$ XØR (B); opération logique "OU exclusif" entre le contenu du registre A et le contenu du registre B (ce dernier contenant le nouvel opérateur $Ø_{A(n)}$), le résultat de cette opération donne $K_{A(n)}$.

10D1 LMA : chargement de $K_{A(n)}$, contenu dans le registre A, en mémoire à la même adresse 10E1, de manière à la retrouver au moment du calcul de sa prochaine valeur.

3 - CALCUL DE LA NOUVELLE VALEUR DE LA SORTIE $S_{A(n)}$:

$S_{A(n)} \leftarrow [RRC(S_{A(n-1)})]$ XØR $(Ø_{A(n)})$

10D2 INL : incrémentation du contenu du registre L permettant de se placer à l'adresse 10E2, où la mémoire contient la valeur $S_{A(n)}$ maintenant considérée comme $S_{A(n-1)}$, la valeur du compteur de temporisation k_A ayant atteint 00.

10D3 LAM : $(A) \leftarrow S_{A(n-1)}$; chargement du contenu de la mémoire à l'adresse 10E2 dans le registre A.

10D4 RRC : rotation à droite du contenu du registre A (chaque bit du registre A se déplace vers la droite, le plus à droite se retrouve le plus à gauche).

10D5 XRB : $(A) \leftarrow (A)$XØR(B); opération logique "OU exclusif" entre le contenu du registre A et le contenu du registre B (ce dernier contenant le nouvel opérateur $Ø_{A(n)}$), le résultat de cette opération donne $S_{A(n)}$.

10D6 LMA : chargement de $S_{A(n)}$, contenu dans le registre A, en mémoire à la même adresse 10E2, de manière à la retrouver au moment du calcul de sa prochaine valeur.

4 - FILTRAGE EVENTUEL DE LA VALEUR DE LA SORTIE AVANT SA SORTIE EFFECTIVE:

10D7 INI 05 : la valeur affichée sur le jeu de clés "05" est immédiatement affectée dans le registre A.

10D8 NDM : $(A) \leftarrow (A)$ AND (M); opération logique "ET" entre le contenu du registre A (i.e.la valeur affichée sur le jeu de clés "05") et le contenu de la mémoire à l'adresse 10E2 (i.e. la valeur de la sortie $S_{A(n)}$), le résultat de cette opération place dans le registre A la valeur de la sortie filtrée.

10D9 ØUT 0B : envoi de l'information, contenue dans le registre A, dans la voie 0B.

5 - POSITIONNEMENT DU COMPTEUR DE TEMPORISATION EN MEMOIRE:

10DA DCL : décrémentation du contenu du registre L permettant de se placer à l'adresse 10E1, où la mémoire contient la valeur du compteur $K_{A(n)}$.

10DB LBM : $(B) \leftarrow K_{A(n)}$; chargement du contenu de la mémoire à l'adresse 10E1 dans le registre B.

10DC DCL }
10DD DCL } : double décrémentation du contenu du registre L permettant de se placer à l'adresse 10DF, où la mémoire contient la valeur du compteur de temporisation $k_{A(n-1)}$; c'est dans le sous-programme de temporisation annexe que nous aurons LMB qui placera la nouvelle valeur du compteur de temporisation $k_{A(n)}$ en mémoire à la même adresse 10DF.

10DE RET : retour au sous-programme de sortie.

IV. SOUS-PROGRAMME DE TEMPORISATION ANNEXE

10FF LMB : chargement du contenu du registre B en mémoire à l'adresse 10DF, le registre B contient soit:

.$K_{A(n)}$: si l'on vient de sortir de la séquence de calcul des nouvelles valeurs associées au paramètre "A"; il s'agit donc du chargement de la nouvelle valeur du compteur de temporisation $k_{A(n)}$.

.la valeur courante k_A venant d'être prélevée dans le programme principal et décrémentée par l'intermédiaire du registre B puis replacée à l'adresse 10DF.

1100 INI 04 : la valeur affichée sur le jeu de clés "04" est immédiatement affectée dans le registre A.

1101 SUI 01 : "décrémentation" du contenu du registre A.

1103 JFZ = 1101 : branchement conditionnel à l'adresse 1101 pour "perdre du temps en machine" et prolonger la temporisation:
si (A) = 00, alors passage à l'instruction suivante (et sortie du sous-programme de temporisation annexe),
si (A) ≠ 00, alors saut à l'adresse 1101.

1106 RET : retour au programme principal.

Remarque: l'instruction SUI 01 placée derrière INI 04 implique que la temporisation annexe la plus courte sera produite en plaçant la valeur 01 sur le jeu de clés "04".

CONSEILS PRATIQUES POUR UNE AUDITION REVEE

- Relier tous les éléments du système hybride au secteur.
- Interconnecter ceux-ci comme il se doit.
- Préparer le "patch" sur le synthétiseur suivant l'une des combinaisons choisie parmi les six proposées.
- Mettre en route l'INTELLEC 8008 et la télétype.
- Entrer le ruban "PARAM" en machine.
- Lancer le programme "COOL" : .G1000
- Lancer le programme "AMBITUS": .G1015
- Pour chaque paramètre, régler la tension moyenne de contrôle sur le synthétiseur, puis son ambitus à l'aide du potentiomètre correspondant sur le 8-DAC; enfin, stopper le programme "AMBITUS".
- Initialiser les valeurs associées aux huit paramètres de 10DF à 10FE.
- Placer 00 sur le jeu de clés "04" et "05".
- Lancer le programme "PARAM" : .G1036
- Modifier progressivement les valeurs affichées sur les jeux de clés.

POUR PASSER DE PARAM A DODECAPARAM :

Dans le cas où l'on désire obtenir des variations dodécaphoniques, il faut accorder le 8-DAC sur le tempérament (dur! dur!):

- Stopper le déroulement de "PARAM".
- Lancer le programme "AMBITUS": .G1015
- Régler le potentiomètre de chaque DAC à accorder jusqu'à entendre un intervalle de deux octaves entre les deux notes délivrées par la sortie.
- Stopper le programme "AMBITUS".
- Lancer le programme "PARAM" : .G1036
- Placer la valeur 01 sur le jeu de clés "04".
- Placer la valeur E0 sur le jeu de clés "05".

On doit alors entendre les notes relatives à un accord dit de quinte augmentée (ex. DO, MI, SOL#).

- En plaçant la valeur F8 sur le jeu de clés "05", on obtient des variations dodécaphoniques pseudo-aléatoires (notes de la gamme chromatique).

Bonnes dodécalcomanies!

Brouillons successifs de Sade réécrivant le Petit Chaperon Rouge

Bernard Leclerc

Artinfo/Musinfo n°30
1980



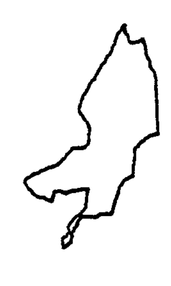
Le programme METAMORPHOSE



Dès que l'ordinateur put s'exprimer par le truchement d'une table traçante , nombreux furent ceux qui pensèrent utiliser le couple ordinateur-traceur pour se livrer à diverses recherches esthético-graphiques ou, plus précisement, donner formes à une combinatoire obsessionnelle et fantasmatique que la manipulation du seul crayon ne laissait déboucher que sur l'anthropomorphe.



C'est ainsi que, vers les années 55, une équipe de jeunes créateurs japonais présentait une composition intitulée RUNNING COLA IS AFRICA où l'on voyait au centre de la feuille le contour d'une bouteille de Coca-cola, flanquée à gauche de la silhouette d'un coureur et à droite du tracé sommaire mais familier du continent africain. Entre le coureur et la bouteille flottait une forme hybride, ligne à peine brisée reliant cent ou deux cents points successifs, milieux respectifs des segments dont les extrémités étaient régulièrement prises sur le coureur et la bouteille; une autre forme hybride flottait entre la bouteille et l'Afrique, traduisant le passage d'une forme à l'autre. D'ailleurs l'ordinateur fournissait d'autres étapes intermédiares, et c'est par une vingtaine de métamorphoses évoluant lentement que l'on passait du coureur à l'Afrique par l'intermédiare de la bouteille.

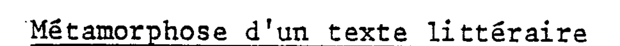


Quinze, vingt ans plus tard, Peter Foldes, utilisant ce procédé de façon systématique, donnait d'étourdissants chefs d'oeuvre du cinéma d'animation, tels que METADATA et HUNGER (ce sont les deux que je connais).



Formulons le procédé, un élément s décrivant la source S tandis qu'un élément o décrit la cible C, on définit un élément h (exactement médian entre s et c, ou si on le désire, voisin soit de l'un soit de l'autre) qui engendre l'hybride H.

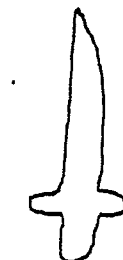
Métamorphose d'un texte littéraire



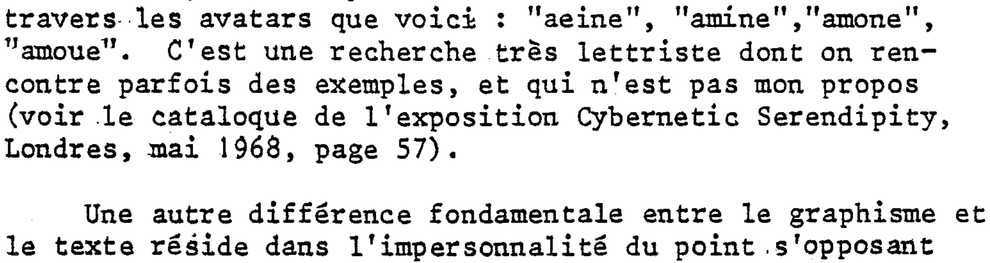


La tentation est forte d'utiliser la même méthode pour passer d'un texte littéraire à un autre. Y succomber n'interdit pas diverses remarques.

Tout d'abord un mot n'est pas un point : si on peut choisir deux points et obtenir sans difficulté le milieu du segment qui les joint, on ne peut définir le mot médian de deux autres. Il faudra donc procéder par tout ou rien, et l'élément intermédiare sera soit le mot source, soit le mot cible.







On pourrait évidemment déchiqueter les mots par lettres successives, le mot "peine" devenant le mot "amour" à travers les avatars que voici : "aeine", "amine","amone", "amoue". C'est une recherche très lettriste dont on rencontre parfois des exemples, et qui n'est pas mon propos (voir le catalogue de l'exposition Cybernetic Serendipity, Londres, mai 1968, page 57).

Une autre différence fondamentale entre le graphisme et le texte réside dans l'impersonnalité du point s'opposant au poids du mot. De la suite de points impersonnels naît l'hybride, tandis que le texte vit à la fois par son ensemble et par ses éléments, par son esprit et par ses mots, ces derniers étant loin d'être impersonnels.



Enfin pour ceux qui prendront connaissance des productions tant graphiques que verbales, interviendra sans doute le fait qu'une image visuelle est perçue plus vite qu'un texte dit ou écrit, et que l'écart par rapport à la moyenne de l'effet obtenu est plus étroit dans le cas du graphisme que dans celui du texte, pour lequel jouent la culture de l'observateur, ses possibilités d'association, son sens de l'humour, l'intensité de ses émotions, etc. Thèse qu'il y aurait intérêt à approfondir.



CHOIX DES TEXTES S ET C

Désireux de voir nettement la métamorphose opérer, j'ai choisi deux domaines littéraires bien différents, l'un noir l'autre rose.

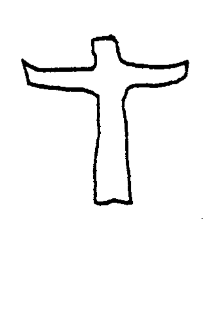
Le texte cible C est un fragment de Sade, choisi dans "Les amours de Saintville et de Lénore" . Choisi au hasard -plus exactement en feuilletant l'ouvrage vers le milieu et en posant le regard sur le premier passage venu dont j'ai su tout de suite que je le retiendrais.

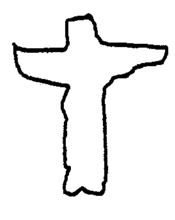
Le texte source S est un condensé du Petit Chaperon Rouge, que j'ai rédigé de mémoire. Je l'ai découpé en cent segments numérotés de 1 à 100 , et chacun comporte un ou plusieurs mots; la longueur maximum d'un segment est de 16 caractères, taille fixée par l'ordinateur.

J'ai imaginé que Sade, ayant à l'esprit le texte de Perrault (qui n'est pas aussi rose que le croyaient nos parents), le faisait évoluer par retouches successives. Hypothèse fantaisiste, à bien des points de vue, et que traduit le titre du présent article ; mais dans le bouillonnement de mots, de souvenirs, d'appels dont disposait Sade lorsqu'il écrivait, pourquoi le Petit Chaperon Rouge n'aurait il pas tenu quelque place, même minime?



Dans une seconde phase de recherche, d'autres couples (S,C) seront envisagés : Jean-Charles Gaudy et moi avons ainsi déjà prospecté le Cantique des Cantiques et l'Amour Fou d'André Breton afin d'en tirer deux fragments dont





s'épouseraient les dissemblances et les ressemblances sémantiques -car il nous semble que le passage d'un texte à l'autre, pour qu'une certaine intelligibilité demeure, appelle des textes de même structure grammaticale, au sens scolaire du mot. Cela s'avère difficile à trouver: faudrait-il réduire les textes à un schéma syntaxique très simplifié, une demi-douzaine de syntagmes par phrase, et prospecter un peu partout dans l'espoir de trouver deux exemples d'une telle structure?



Ces réflexions nous ont conduit à trois autres couples (S,C). Le premier utilise comme source S un texte de Marx - quelques lignes du "Capital" - qui commence ainsi: "Le travail est de prime abord un acte qui se passe entre l'homme et la nature...", et comme cible C un texte qui, ne s'étant pas présenté au cours de notre prospection de la Bible, a été créé en respectant assez fidèlement la structure grammaticale de S. Voici sa première phrase : "L'amour sera premièrement le lien qui unira le Juste et la vierge ...".



Le second couple utilise deux textes du même ouvrage de René THOM, "Modèles mathématiques de la morphogenèse", collection 10-18.

Le troisième couple est constitué par deux productions que nous avons écrites indépendamment l'un de l'autre : l'une est poétique, l'autre est mathématique, et chacune comporte quatre phrases, les deux premières en substantifs tous masculins, les deux dernières tous féminins. Chaque couple de phrases aura comme structure, avec des notations évidentes : ADV GN+ADJ V GN(CI) "et" V ADV GN(CI) "." ADVGN "et" GN V GN(cd) "," CONJ GN V GN(CI) "." .

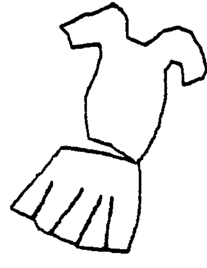


ECRITURE DU PROGRAMME

Le petit ordinateur WANG 2000 que j'ai utilisé est programmé en langage BASIC. Une dizaine de lignes d'instructions suffit au programme "PEROSADE" proprement dit, qui fait suite à la longue introduction des données(DATA).

Il y a quatre textes intermédiares, et , pour chacun d'eux, à tour de rôle, deux segments T1& de chaque dizaine de segments du texte source sont transformés en blancs, " "; les segments de même indice du texte cible T2& sont transportés dans le texte intermédiaire T3& par l'instruction T3&(L)=T2&(L), puis transformés en blancs (T2&(L)=" "). On imprime alors tous les segments non blancs du texte intermédiaire (PRINT T3&(M)), et chaque fois qu'un segment blanc se présente, on le remplace par le segment correspondant du texte source T1&(M), qui évidemment n'est pas vide. Et les illustrations (programme CHAPLOUP) procèdent de la méthode initiale.



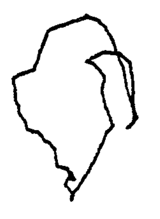




Le numéro d'ordre des segments modifiés, dans chaque dizaine, est déterminé par une procédure pseudo-aléatoire (Random), câblée dans l'ordinateur (RND(1)). L'aléatoire, du reste, n'est pas de rigueur - c'est d'ailleurs, comme tout ce que calcule la machine, un pseudo-aléatoire, c'est à-dire que les calculs qui l'engendrent brouillent assez les traces pour qu'on ne retrouve plus une régularité trop évidente. D'autres procédures, plus régulières, déchiquèteraient aussi bien les significations.

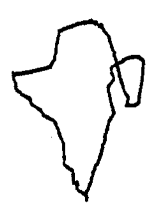






BROUILLONS SUCCESSIFS DE SADE
REECRIVANT LE CHAPERON ROUGE

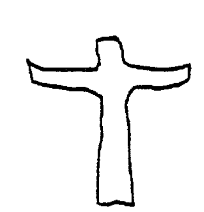
POUR ALLER VOIR SA MERE GRAND CHAPERON ROUGE MIT DANS SON PANIER NOIX-DE-CAJOU WHISKY KLEENEX PARIS-MATCH GALETTE ET POT-DE-BEURRE. LA VOILA PARTIE A TRAVERS LA FORET CUEILLANT NOISETTE COURSANT PAPILLON LIANT FLEURETTES EN BOUQUETS BELLE COMME UN COEUR. PROMENONS-NOUS DANS LES BOIS PENDANT QUE LE LOUP N'Y EST PAS,CHANTENT LES FILLETTES DANS LES PRES. NOUS N'IRONS PLUS AU BOIS, CHANTENT LES AMANTS DELAISSES.DANS LE LIT, LE LOUP FREMISSAIT D'AISE EN SENTANT LA CHAIR FRAICHE SE FROTTER CONTRE SON POIL RUDE. 'MERE GRAND QUE VOUS AVEZ DE GRANDS YEUX.' 'JE VAIS TE FAIRE TA FETE MON ENFANT.' LE LOUP L'ETREIGNIT AVEC VIOLENCE ET SE LA FARCIT SANS PLUS D'HISTOIRES.

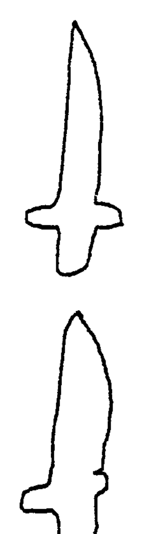


POUR L'AYANT VOIR SA MERE GRAND SON ROUGE MIT DANS SON VICAIRE NOIX-DE-CAJOU WHISKY TOUS PARIS-MATCH GALETTE ET POT-DE-BEURRE. LA VOILA GRAND A TRAVERS LA FORET CUEILLANT NOISETTE PARTIES PAPILLON LIANT FLEURETTES EN BOUQUETS FILLE, COMME UN COEUR. PROMENONS-NOUS DANS LES BOIS PENDANT QUE LE LOUP N'Y BOUT D'UN ASSEZ PAS,CHANTENT ON FILLETTES DANS LES PRES. NOUS EN PLUS AU BOIS, CHANTENT LES CE DELAISSES.DANS LE



LIT, LE LOUP LE D'AISE ETAIT SENTANT LA CHAIR FRAICHE SE FROTTER SOIN SON PLUS RUDE. LES GRAND QUE VOUS AVEZ DE GRANDS YEUX.' PAS VAIS TE FAIRE RECHERCHES FETE MON ENFANT.' LE LOUP L'ETREIGNIT AU VIOLENCE ET SE LA FARCIT SANS NOIR PRESQUE D'HISTOIRES.





POUR L'AYANT VOIR SA MERE GRAND SON FAUTEUIL MIT ENTRE LE SON VICAIRE NOIX-DE-CAJOU WHISKY TOUS PARIS-MATCH GALETTE ET LE PLUS GRAND SOIN FORET CUEILLANT NOISETTE PARTIES PAPILLON LIANT DE EN BOUQUETS FILLE, OUI PROMENONS-NOUS DANS LES BOIS PENDANT QUE LE LOUP N'Y BOUT D'UN ASSEZ LONG,CHANTENT ON FILLETTES DANS LES CHANGER NOUS EN PLUS QU'ELLE BOIS, CHANTENT A L'UN CE QU'ELLE.DANS LE LIT, LE LOUP LE D'AISE ETAIT PROFOND; ON CHAIR FRAICHE SE FROTTER SOIN SON PLUS RUDE. LES GRAND VERIFIAIENT CE AVEZ DE GRANDS YEUX.' PAS VAIS ET FACILITAIENT FAIRE RECHERCHES FETE MON ENFANT.' LE L'INQUISITEUR L'ETREIGNIT AU VIOLENCE DU SE LA UN SANS NOIR PRESQUE D'HISTOIRES.



POUR L'AYANT VOIR SA MOUVEMENT GRAND SON FAUTEUIL ENFERMEE ENTRE LE SON VICAIRE NOIX-DE-CAJOU LUI TOUS PARIS-MATCH GALETTE AVEC LE PLUS GRAND SOIN CHACUNE CUEILLANT NOISETTE PARTIES PAPILLON CORPS DE EN BOUQUETS FILLE, OUI PROMENONS-NOUS DANS LES BOIS PENDANT VERS LE LOUP AU BOUT D'UN ASSEZ LONG TEMPS ON FILLETTES DANS LES CHANGER NOUS EN SORTE QU'ELLE BOIS, CHANTENT A L'UN CE QU'ELLE.DANS DE LIT, LE L'AUTRE. LE D'AISE ETAIT PROFOND; ON CHAIR DE FORT PRES ET FROTTER SOIN SON PLUS RUDE. LES GRAND VERIFIAIENT CE AVEZ DE NE DISCERNAIT PAS BIEN ET FACILITAIENT FAIRE RECHERCHES FETE MON LES LE L'INQUISITEUR OBSERVA AU VIOLENCE DU SE LA UN SANS NOIR PRESQUE IMPERCEPTIBLE.



POUR L'AYANT PAR SA MOUVEMENT DE SON FAUTEUIL ENFERMEE ENTRE LE SON VICAIRE ET LUI TOUS DEUX GALETTE AVEC LE PLUS GRAND SOIN CHACUNE CUEILLANT DIFFERENTES PARTIES PAPILLON CORPS DE CETTE FILLE, OUI PROMENONS-NOUS DANS LES BOIS TOURNEE VERS EUX. AU BOUT D'UN ASSEZ LONG TEMPS ON LA FIT CHANGER NOUS EN SORTE QU'ELLE BOIS, CHANTENT A L'UN CE QU'ELLE VENAIT DE PRESENTER LE L'AUTRE. LE D'AISE ETAIT PROFOND; ON CHAIR DE FORT PRES ET AVEC LE SOIN LE PLUS RUDE. LES GRAND VERIFIAIENT CE QUE L'OEIL NE DISCERNAIT PAS BIEN ET FACILITAIENT LES RECHERCHES FETE FIXAIENT LES LE L'INQUISITEUR OBSERVA AU BORD DU SE DROIT UN SANS NOIR PRESQUE IMPERCEPTIBLE.



DOM CRISPE L'AYANT PAR UN MOUVEMENT DE SON FAUTEUIL ENFERMEE ENTRE LE GRAND VICAIRE ET LUI TOUS DEUX EXAMINERENT AVEC LE PLUS GRAND SOIN CHACUNE DES DIFFERENTES PARTIES DU CORPS DE CETTE FILLE, QUI SE TROUVAIT TOURNEE VERS EUX. AU BOUT D'UN ASSEZ LONG TEMPS ON LA FIT CHANGER D'ATTITUDE EN SORTE QU'ELLE OFFRAIT MAINTENANT A L'UN CE QU'ELLE VENAIT DE PRESENTER A L'AUTRE. LE SILENCE ETAIT PROFOND; ON OBSERVAIT DE FORT PRES ET AVEC LE SOIN LE PLUS EXACT LES DOIGTS VERIFIAIENT CE QUE L'OEIL NE DISCERNAIT PAS BIEN ET FACILITAIENT LES RECHERCHES OU FIXAIENT LES POSITIONS. L'INQUISITEUR OBSERVA AU BORD DU SEIN DROIT UN SIGNE NOIR PRESQUE IMPERCEPTIBLE.



```
10 REM "PEROSADE"
20 PRINT "                    BROUILLONS SUCCESSIFS DE SADE"
30 PRINT "                         REECRIVANT LE CHAPERON ROUGE"
40 PRINT
50 DIM T1$(100),T2$(100),T3$(100)
60 FOR I=1 TO 100:READ T1$(I):PRINT T1$(I);:NEXT I
70 PRINT :PRINT
80 FOR I=1 TO 100:READ T2$(I):NEXT I
90 DATA "  POUR"," ALLER"," VOIR"," SA"," MERE"
100 DATA " GRAND"," CHAPERON"," ROUGE"
110 DATA " MIT"," DANS"," SON"," PANIER"," NOIX-DE-CAJOU"
120 DATA " WHISKY"," KLEENEX"," PARIS-MATCH"
130 DATA " GALETTE"
140 DATA " ET"," POT-DE-BEURRE."," LA VOILA"," PARTIE"
150 DATA " A TRAVERS LA"," FORET"," CUEILLANT"," NOISETTE"
160 DATA " COURSANT"," PAPILLON"," LIANT"," FLEURETTES"
170 DATA " EN BOUQUETS"," BELLE"," COMME UN COEUR."
180 DATA " PROMENONS-NOUS"," DANS LES BOIS"
190 DATA " PENDANT"," QUE"," LE LOUP"," N'Y"," EST"," PAS"
200 DATA ",CHANTENT"," LES"," FILLETTES"," DANS LES"," PRES."
210 DATA " NOUS"," N'IRONS"," PLUS"," AU"," BOIS,"
220 DATA " CHANTENT"," LES"," AMANTS"," DELAISSES"
230 DATA ".DANS"," LE"," LIT,"," LE"," LOUP"," FREMISSAIT"
240 DATA " D'AISE"," EN"," SENTANT"," LA"," CHAIR"
250 DATA " FRAICHE"," SE"," FROTTER"," CONTRE"," SON"
260 DATA " POIL"," RUDE."," 'MERE"," GRAND"
270 DATA " QUE"," VOUS"," AVEZ"," DE"
280 DATA " GRANDS"
290 DATA " YEUX.'"," 'JE"," VAIS"," TE"," FAIRE"," TA"
300 DATA " FETE"," MON"," ENFANT.'"," LE"," LOUP"
310 DATA " L'ETREIGNIT"
320 DATA " AVEC"," VIOLENCE"," ET"," SE"," LA"," FARCIT"
330 DATA " SANS"," PLUS"," D'HISTOIRES."
340 DATA " DON CRISPE"," L'AYANT"," PAR"," UN"," MOUVEMENT"
350 DATA " DE"," SON"," FAUTEUIL"," ENFERMEE"," ENTRE LE"
360 DATA " GRAND"
370 DATA " VICAIRE"," ET"," LUI"," TOUS"," DEUX"
380 DATA " EXAMINERENT"," AVEC"," LE"," PLUS"
390 DATA " GRAND"
400 DATA " SOIN"," CHACUNE"," DES"," DIFFERENTES"," PARTIES"
410 DATA " DU"," CORPS"," DE"," CETTE"," FILLE,"," OUI"," SE"
420 DATA " TROUVAIT"," TOURNEE"," VERS"," EUX."," AU"
430 DATA " BOUT D'UN ASSEZ"," LONG"
440 DATA " TEMPS"," ON"," LA"," FIT"
450 DATA " CHANGER"
460 DATA " D'ATTITUDE"," EN"," SORTE"," QU'ELLE"," OFFRAIT"
470 DATA " MAINTENANT"," A L'UN"," CE"," QU'ELLE"
480 DATA " VENAIT"," DE"," PRESENTER"," A"," L'AUTRE."
490 DATA " LE"," SILENCE"," ETAIT"," PROFOND;"," ON"
500 DATA " OBSERVAIT"," DE FORT PRES"," ET"," AVEC LE"
510 DATA " SOIN"," LE"," PLUS"," EXACT"," LES"
520 DATA " DOIGTS"
530 DATA " VERIFIAIENT"," CE"," QUE"," L'OEIL"," NE"
540 DATA " DISCERNAIT"," PAS"," RIEN"," ET FACILITAIENT"
550 DATA " LES"," RECHERCHES"," OU"," FIXAIENT"," LES"
560 DATA " POSITIONS."," L'INQUISITEUR"," OBSERVA"," AU"
570 DATA " BORD"," DU"," SEIN"," DROIT"," UN"
580 DATA " SIGNE"," NOIR PRESQUE"," IMPERCEPTIBLE."
590 FOR A=0 TO 4:FOR I=0 TO 9:J=1
600 L=10*I+1+10*RND(1):IF T1$(L)=" " THEN 600
610 T1$(L)=" ":T3$(L)=T2$(L):T2$(L)=" ":J=J+1
620 IF J=2 THEN 600
630 FOR N=1 TO 10:M=10*I+N:IF T2$(M)=" " THEN 650
640 PRINT T1$(M);:GOTO 660
650 PRINT T3$(M);
660 NEXT N:NEXT I:PRINT :PRINT :NEXT A
670 END
```

```
10 REM CHAPLOUP MARDI 25 MARS 1980
20 DEFFN'0"PLOT["
30 DIM S(5,60),T(5,60)
40 REM 2 FOIS 60 DATA POUR COUTEAU,LOUP,CHAP,CHRIST,GDMR
50 DATA 29,138,32,134,36,126,40,117,42,108,43,100,44,92
60 DATA 44,83,42,74,41,52,39,43,40,41,48,40,54,37,55,30
70 DATA 51,29,47,28,43,28,40,29,39,27,38,23,37,20,36,17
80 DATA 36,14,35,11,34,8,32,6,30,5,28,4,25,3,20,4,18,6
90 DATA 18,10,19,15,19,20,20,23,20,26,18,29,13,29,10,29
100 DATA 8,30,4,30,3,31,1,32,1,33,1,36,2,38,5,39,10,40
110 DATA 15,40,18,42,20,46,21,50,22,60,23,70,24,80,25,90
120 DATA 28,130,29,137,29,138
130 DATA 88,115,93,99,104,113,107,93,112,90,112,38
140 DATA 105,43,96,48,85,48,82,40,73,40,70,38,58,30,50,27
150 DATA 45,22,37,22,36,19,34,21,29,20,32,23,30,26,24,22
160 DATA 20,12,18,7,14,6,12,8,13,15,18,23,25,29,30,30,32,35
170 DATA 34,32,39,36,50,40,55,40,56,57,33,50,32,45
180 DATA 30,49,26,46,25,36,20,42,15,41,12,43,9,47,4,48
190 DATA 2,49,3,52,5,56,28,68,42,82,51,77,56,80,57,84
200 DATA 51,81,42,82,47,90,78,100,83,108,88,115
210 DATA 60,168,67,167,70,162,71,148,83,143,90,148,97,153
220 DATA 105,155,115,150,122,140,125,128,122,120,108,115
230 DATA 108,124,106,131,101,133,96,127,95,110,90,90
240 DATA 85,80,81,72,75,66,77,57,76,42,74,31,72,20,66,5
250 DATA 52,7,57,42,49,7,30,9,49,43,30,10,13,18,37,48
260 DATA 12,18,4,22,0,28,15,53,0,28,-15,36,23,75,53,70
270 DATA 74,64,56,77,40,82,39,92,39,108,48,124,52,137
280 DATA 50,142,49,144,38,143,30,142,28,150,23,157
290 DATA 35,162,41,164,48,166,60,168
300 DATA 50,97,52,96,54,94,55,88,56,84,59,82,64,81
310 DATA 73,80,79,80,86,81,93,83,92,80,91,75,90,74
320 DATA 86,72,81,71,75,70,72,70,70,71,62,71,60,70
330 DATA 58,68,56,63,58,52,58,40,58,32,59,24,59,16
340 DATA 60,8,57,8,50,10,45,8,42,8,39,9,40,20,40,26
350 DATA 41,35,43,45,42,50,43,56,43,60,42,66,38,68
360 DATA 33,70,24,70,20,71,12,72,7,73,4,74,1,86,6,82
370 DATA 10,80,20,80,40,80,42,83,42,87,40,89,40,96
380 DATA 44,97,50,97
390 DATA 63,116,77,111,90,93,100,69,106,42,110,21
400 DATA 93,12,89,13,86,20,79,50,68,80,58,94,43,100
410 DATA 36,98,26,92,34,93,46,92,57,84,63,76,68,65
420 DATA 76,43,86,27,78,24,70,21,60,13,47,4,40,3,34,4
430 DATA 36,8,40,14,38,21,51,38,57,30,52,31,49,30,37,27
440 DATA 34,37,25,34,24,30,18,29,14,30,12,24,8,26,10,31
450 DATA 13,39,18,46,22,49,27,53,30,56,30,58,26,61,22,68
460 DATA 19,71,18,78,19,86,22,90,26,92,34,103,55,113,63,116
470 FOR I=1 TO 5:FOR J=1 TO 60:READ S(I,J)
480 READ T(I,J):NEXT J:NEXT I
490 FOR J=1 TO 60:S(1,J)=S(1,J)+59:T(1,J)=T(1,J)-23
500 S(3,J)=40+INT(.8*S(3,J)):T(3,J)=INT(.8*T(3,J))-21
510 S(4,J)=S(4,J)+38
520 T(4,J)=T(4,J)+18:S(5,J)=S(5,J)+25:T(5,J)=T(5,J)-1
530 NEXT J:FOR L=1 TO 5
540 PLOT [,,R],[140,1075-175*L,]
550 IF L[]3 THEN 560:PLOT [,-25,]
560 IF L=5 THEN 570:M=L+1:GOTO 580
570 M=1:PLOT [,35,]:GOTO 580
580 IF L[]4 THEN 590:PLOT [,-29,]
590 FOR D=0 TO 6:X0=S(1,1):Y0=T(1,1)
600 FOR J=2 TO 59
610 X=S(L,J)+INT(D*(S(M,J)-S(L,J))/6)
620 Y=T(L,J)+INT(D*(T(M,J)-T(L,J))/6)
630 PLOT [Y-Y0,X0-X,D]:X0=X:Y0=Y:NEXT J
640 PLOT [T(1,1)-Y0,X0-S(1,1),D]
650 IF D=6 THEN 660:PLOT [155,,]
660 NEXT D
670 NEXT L:PLOT [,,R]
680 END
```

A propos de l'Enseignement de l'Informatique Musicale à l'Université

Jacques Arveiller

Communication présentée à l'UNESCO Workshop on Computer Music,
Institute of Musicology, Université d'Aarhus, Danemark, le 1 Septembre 1978.

Artinfo/Musinfo n°30
1980

Un compositeur de musique se réfère toujours dans sa pratique à d'autres compositeurs, du passé et du présent(1). L'idée-même de composition musicale est indissociable de la question de sa transmission pédagogique, avec tout ce que cela comporte d'aspects psychologiques et institutionnels. Car une partie seulement d'un savoir compositionnel est susceptible d'être écrite, sous forme de traité, de manuel, de méthode.

Avec l'entrée en scène des ordinateurs, si la question de fond reste la même, la didactique de la composition va présenter quelques particularités et va achopper sur quelques difficultés spécifiques.

Dans la mesure où, de plus en plus, l'ordinateur va interférer dans la composition musicale, il m'apparaît profitable de relater ici une expérience qui se déroule depuis 1970, et d'indiquer quelques solutions que nous avons tenté de donner à ces difficultés.

Rappelons tout d'abord pour mémoire quelques particularités de la pratique musicale avec ordinateur. Deux points semblent important dans la mesure où ils font surgir des problèmes nouveaux, justiciables de solutions pédagogiques nouvelles.

D'une part la composition musicale avec ordinateur suppose la pratique simultanée de deux activités créatrices qui se chevauchent, avec le risque constant que l'une prenne le pas sur l'autre. Je considère en effet la programmation comme une activité créatrice très complète, qui peut dans certains cas se suffire à elle-même.

D'autre part on assiste à une fusion des fonctions musicales traditionnelles. Au-delà de l'indistinction compositeur/exécutant telle qu'elle se présente par exemple dans l'improvisation, l'informatique musicale instaure dans bien des cas une indistinction entre composition, exécution et lutherie, ce qui n'est pas sans conséquences au niveau des étudiants.

---

(1) Je tiens à remercier ici Patrick Greussay, qui m'a confié un enseignement dont il avait auparavant la charge. Je lui suis redevable, directement ou indirectement, des quelques idées exprimées dans cette communication.

Un second ordre de particularités est lié à l'implantation universitaire. L'informatique musicale ne peut en effet se pratiquer que là où se trouvent des ordinateurs, et là où l'on est prêt à mettre en jeu ces ordinateurs pour faire de la musique. La musique réalisée avec ordinateur est encore une des plus chères qui soient, du point de vue du coût de production. C'est dire que pour le débutant, le lieu privilégié de l'apprentissage est actuellement encore l'université.

L'Université de Paris 8-Vincennes regroupe 32500 étudiants dont 900 en Musique et 1700 en Informatique. Les cours d'informatique musicale sont suivis principalement par les étudiants de ces deux départements, auxquels s'ajoutent des étudiants venus d'autres départements, et un certain nombre d'auditeurs libres (principalement musiciens professionnels venus apprendre pour leur propre compte une technique nouvelle). L'ouverture des cours à toute personne désireuse d'y assister a donc pour conséquence une grande hétérogénéité au niveau du bagage préalable.

Vincennes est une université à vocation plus littéraire que scientifique, donc munie de moyens de calcul relativement limités (Téléméca-nique T1600 et terminal de PDP10 en télétraitement). L'accent y est donc mis d'une part sur la construction de *software*, d'autre part sur les microprocesseurs et leurs applications. Parmi les réalisations récentes orientées sur les applications artistiques à la disposition des étudiants, on note ainsi par exemple un système LISP sur microprocesseur (VLISP 8 de J. Chailloux), une télévision en couleurs gérée point par point par microprocesseur (COLORIX de L. Audoire) ou le Système Portable de Synthèse Hybride permettant le contrôle de synthétiseurs de son (fruit d'une collaboration : D. Roncin, C. Colère, P. Greussay, etc.).

L'Université de Vincennes a été créée dans une optique pluridisciplinaire. Par voie de conséquence s'y est toujours développée une collaboration de tous les instants entre les différents départements (Informatique, Musique, Arts Plastiques). Et du même coup un travail collectif entre musiciens et artistes plasticiens.
Il faut dire que le Département d'Informatique lui-même a été créé en 1969 par des peintres et musiciens à la recherche d'un nouvel outil de création. L'optique vincennoise est donc un peu particulière : si l'on enseigne toutes les branches de l'informatique, l'idée règne que faire du son ou des images est une des meilleures choses qu'on puisse faire avec un ordinateur, et non pas une activité un peu superflue ou vaguement honteuse. Les machines y sont donc en libre-service, et pas seulement à certaines heures.

Comme dernière particularité de Vincennes, il faut aussi signaler l'existence du Groupe Art et Informatique de Vincennes (GAIV). Ouvert sur l'extérieur, puisqu'il s'occupe de la production de concerts, d'expositions, de spectacles et de conférences, il regroupe des artistes qui sont soit enseignants soit étudiants. Son articulation avec l'université est souple : indépendance sur le plan administratif, préservant cependant la possibilité d'utilisation extérieure d'un matériel appartenant à l'université.
Une université est par essence un lieu de passage. Le GAIV en est un lui aussi. Il ne tire pas sa cohérence d'une direction d'ensemble ou d'un projet collectif (auquel il conviendrait de se soumettre) mais d'une référence commune à un même outil technologique

(par exemple le microprocesseur) et à une même méthodologie (comme la référence à l'Intelligence Artificielle).

Si le *software* est le plus souvent commun, la trace musicale est toujours personnelle, et l'on s'y défie de l'orthodoxie esthétique. Les musiques produites avec l'ordinateur à Vincennes sont donc très diverses dans leur style et leurs références.

★
★ ★

Un troisième et dernier ordre de particularité est lié à la pédagogie. Trois cours annuels (neuf heures par semaine) sont spécifiquement dévolus à l'informatique musicale : Musique et Informatique (J. Arveiller), Synthèse Musicale Hybride (G. Dalmasso) et Formalisation Informatique de Structures Musicales (A. Riotte). S'y ajoutent la part informatique de certains enseignements de musique (M. Battier, G.G. Englert) et la part de la musique dans certains enseignements d'informatique (P. Greussay). On n'oublie pas non plus ce qu'apportent les conférenciers extérieurs.

Sans vouloir subsumer la diversité vincennoise, les lignes qui suivent sont donc plutôt des réflexions d'ordre personnel sur la façon dont un enseignant parmi d'autres envisage sa position.

L'enseignement de l'informatique musicale me semble caractérisé par un style pédagogique assez particulier. Dans l'enseignement scientifique traditionnel, la démarche d'apprentissage apparaît par bien des aspects identique à la démarche de recherche scientifique, utilisant les mêmes voies, et demandant le même type d'efforts. A ceci près : on ne demande pas à l'étudiant de trouver, mais de re-trouver ce que d'autres ont déjà découvert avant lui. Le but du chercheur scientifique est clair : montrer à d'autres chercheurs un résultat original, obtenu souvent avec du matériel de laboratoire perfectionné. L'effort de l'étudiant - comme celui de l'enfant - débouchera sur tout autre chose : il montrera par exemple à un enseignant qu'il sait démontrer un théorème. Mais dans tout cela, ni collègues, ni matériel, ni surtout de plaisir. L'enseignement scientifique réclame trop souvent chez l'étudiant l'effort préparatoire au plaisir, tout en l'excluant du plaisir de trouver.

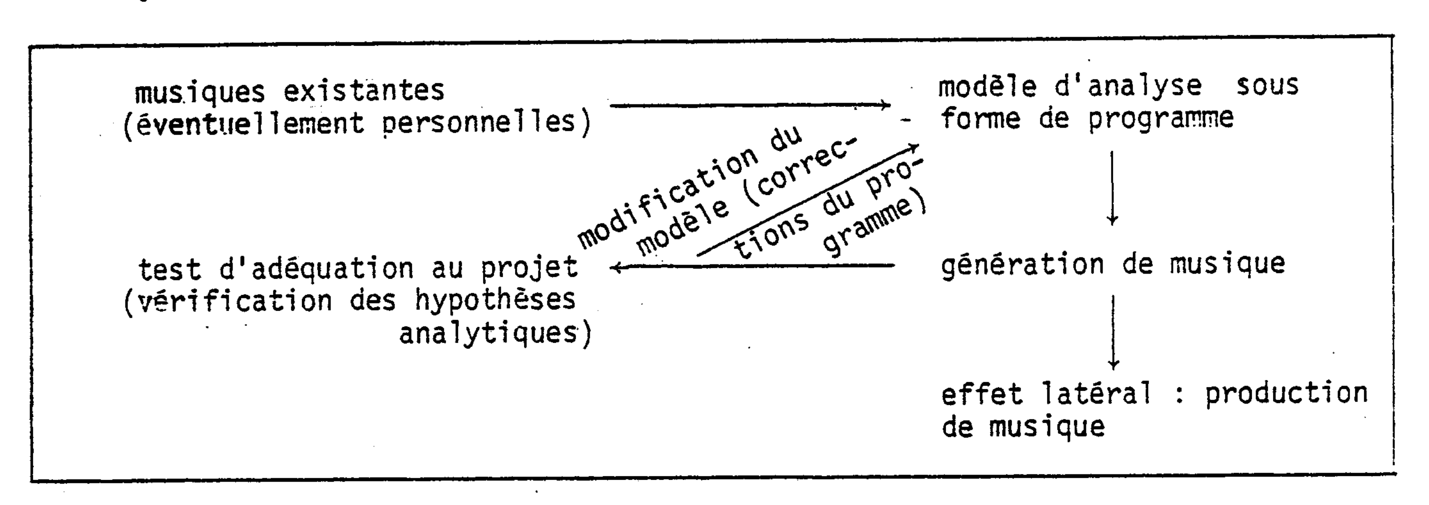
L'utilisation de périphériques musicaux ou visuels permet de donner une appréhension certes partielle, mais en tout cas intuitive, immédiate, sensible de concepts abstraits transmis habituellement dans l'enseignement des mathématiques, de la logique ou de la physique. La maîtrise d'un concept nouveau y renvoie par conséquent d'abord à un plaisir, avant de renvoyer à l'institution enseignante. C'est à partir de ce plaisir que l'étudiant désirera maîtriser d'autres concepts.

Le style de pédagogie que nous développons vise donc à favoriser un mode de communication entre étudiants proche de celui qui lie deux chercheurs. Un glissement comparable semble modifier aussi le rapport maître-élève.

Dans un contexte logico-mathématique habituel, il est normal de penser qu'il existe de bonnes solutions à un problème, et que certaines solutions sont meilleures que d'autres. En programmation usuelle, on peut dire qu'en fonction d'un but déterminé d'avance, il existe des programmes plus performants que d'autres. La programmation de compositions musicales est toute différente : il est licite de modifier en cours d'élaboration le but fixé à l'avance, et les compositeurs ne s'en privent pas. Peut-on établir dès lors une hiérarchie des itinéraires, lorsqu'on ne connaît pas d'avance le but du voyage? Les bonnes solutions deviennent caduques. Et lorsqu'elles persistent, elles prennent souvent un tour

largement empirique. On comprend qu'enseignant et enseigné entretiennent volontiers des rapports différents du modèle habituel.
D'autres facteurs viennent encore y concourir : la machine est un juge sans passion pour dire qu'un programme tourne (ce qui permet à l'enseignant de faire l'économie de certaines appréciations subjectives) ; l'enseignant et l'étudiant prennent les mêmes risques en proposant leur musique à des auditeurs ; être compositeur enfin n'est pas un grade universitaire, ce qui entraîne un décentrement des processus identificatoires habituellement mis en jeu à l'université.

Le parcours pédagogique proposé aux étudiants suggère de mêler chaque fois que c'est possible le *faire* et le *se voir faire*. L'informatique a ceci de particulier qu'elle permet une reversibilité immédiate des modèles d'analyse : l'analyse d'une pièce musicale sous forme de programme, aussi approximatif et incomplet que soit le décryptage des processus compositionnels supposés la sous-tendre, permet l'utilisation immédiate de ce programme "en génération", pour confirmation ou infirmation des hypothèses analytiques formulées. Mais dans l'un et l'autre cas de la musique est produite par effet latéral, musique écoutable en fonction de critères autres que ceux qui ont trait à l'activité d'analyse :



Est-il risqué de supposer qu'il puisse exister ainsi un jeu d'allers et retours entre le Vrai et le Beau? Est-il risqué d'avancer qu'une pièce musicale réussie puisse s'originer dans une analyse invalide? Une erreur n'est jamais fortuite. Une erreur dans l'analyse n'est-elle pas, dans certains cas, ce qui justement appartient en propre à l'analyste, et la marque de la subjectivité du compositeur?

Laissant ces questions ouvertes, il faut quand même insister sur la notion de fin. J'aimerais dire qu'un programme de composition n'est jamais fini, que sa réalisation n'est qu'une suite d'états entrecoupés de modifications et de corrections d'erreurs, corrections toujours significatives par rapport à la représentation que se fait un compositeur de sa propre pièce en cours, par rapport à ce qu'il sait de lui-même.

Ceci débouche sur un des points que je considère comme important pour les étudiants : écrire des programmes compositionnels est certes un moyen d'apprendre sur la musique, mais aussi un moyen d'apprendre sur soi-même en train de faire de la musique. J'envisage volontiers les

programmes

à composer (mais aussi : à lire, à improviser) de la musique comme des programmes d'Intelligence Artificielle, et j'essaye de faire travailler les étudiants dans la perspective que composer par programme, c'est toujours pour une part se simuler soi-même en train de composer.

*
* *

Venons-en aux difficultés, en examinant dans un premier temps celles qui sont directement liées à la pratique musicale avec ordinateurs.

Il faut d'abord reconnaître qu'il existe un hiatus entre le style de travail du compositeur et le style de travail du programmeur. Le compositeur traditionnel peut mener un travail isolé, visant à un résultat unique (une pièce particulière). Le processus compositionnel est contrôlé de bout en bout par une seule personne.

Le travail informatique impose une très forte dépendance par rapport à la machine (au niveau des pannes par exemple), par rapport aussi à un certain nombre d'autres personnes de technicités différentes. Mais ce qui est plus important peut être la perte de contrôle par le compositeur du processus qu'il a lui-même engagé : un programme est, on le sait, la définition d'un processus général, aux virtualités telles que tous les résultats possibles ne sont pas explorables exhaustivement ni représentables a priori, susceptible par conséquent de donner dans un cas précis des résultats inattendus ou musicalement inacceptables.

Sauf à adopter la position de principe d'accepter les yeux fermés ce qui advient une fois le programme lancé, cette perte de prévisibilité sur le résultat musical me semble à l'origine de l'échec de certains compositeurs déjà formés dans leurs tentatives de passage à la composition automatique.

Les rapports sont aussi parfois un peu difficiles entre les musiciens et ceux qui les entourent : scientifiques (mathématiciens, physiciens) et techniciens (électroniciens, programmeurs "purs").

On assiste parfois à un dialogue de sourds entre le musicien et le scientifique, celui-ci ayant tendance à indexer un jugement esthétique sur une argumentation touchant à la trivialité mathématique des algorithmes mis en jeu dans la composition. Un musicien le supporte mal : pour lui seul le résultat musical compte.

La collaboration avec le technicien a tendance à prendre dans certains cas un tour vaguement persécutif : sentiment chez le technicien d'avoir fait la plus grande partie du travail sans pouvoir en tirer le bénéfices publics, de voir un autre se contenter de signer pour recueillir les applaudissements. Le technicien, ou réputé tel, peut alors adopter une position ambivalente : réaliser du *hardware* ou du *software* à vocation musicale sans l'utiliser en son propre nom (puisqu'il ne se revendique pas comme artiste et ne se sent pas armé pour le concert) tout en empêchant plus ou moins consciemment de musiciens de l'employer (sa créativité refoulée s'incarnant alors dans des modifications "hard" ou "soft" deux jours avant le concert, pour le plus grand dépit du musicien

De son côté, le musicien pourra se sentir en risque d'être dépossédé de son objet propre à un double niveau : au niveau conceptuel par le scientifique, au niveau technologique par le technicien. Ce sentiment - un peu magique - de dépossession n'est d'ailleurs pas nouveau. On l'avait déjà vu opérer pour la musique électronique, qui témoignait d'une mutation : la lutherie n'était plus affaire artisanale, mais branche privilégiée de l'industrie, avec tout ce que ça comporte de

rentabilisation et d'amortissement commercial des matériels nouveaux.

Pour éviter ces écueils, dont j'exagère sans doute l'importance en dramatisant un peu le propos, j'insiste sur la nécessité que les étudiants en musique acquièrent une relative indépendance : il ne s'agit pas d'être omniscient ou omnicompétent, mais que le niveau informatique du compositeur soit suffisant pour lui permettre d'exprimer avec souplesse et précision des impératifs compositionnels qui lui sont tellement personnels qu'ils risquent de ne pas pouvoir être communiqués à autrui sans malentendu.

Mais les rapports interhumains les plus difficiles sont certainement ceux que l'on entretient avec soi-même. Le travail compositionnel avec ordinateur me semble particulièrement exposé à une certaine fuite en avant : les programmes réalisés ne sont pas communiqués, car on n'a pas le temps de les annoter . Le temps manque pour en donner une version commentée et lisible car on est déjà pris par la réalisation d'un nouveau programme. Résultat : tout un *software* passionnant, fruit d'une dépense d'énergie passionnée, risque de n'être finalement utilisé que par celui qui l'a conçu.

Plus : le programme aux virtualités les plus générales n'est souvent utilisé qu'une seule fois, pour une pièce particulière, par celui qui l'a écrit. Dans certains cas, cela débouche même sur un véritable art conceptuel : le programme musical ne produit pas même une seule fois de la musique. Il a tourné, c'est suffisant, et on passe à un autre programme. On a l'impression par moments que l'effort mis en jeu pour l'élaboration d'un programme est démesuré par rapport à l'exploitation parcimonieuse (ou absente) qui en sera faite.

L'informatique augmente encore chez certains ce risque connu : à force de se décrire soi-même en train de faire de la musique, avec tout ce que cela comporte de complaisance narcissique, on ne fait plus du tout de musique. La fabrication d'un *software* débouche sur la conception d'un autre *software* et la production proprement musicale disparaît au profit de la confection de programmes qui ne seront jamais exploités.

Comment, à l'université, où la fonction de communication doit tenir la première place, et où l'on n'est pas toujours astreint à justifier son travail par des traces écrites, parer au danger d'une activité tendant vers la récursion sans condition d'arrêt ni effet latéral?

Nous insistons à Vincennes, sur le fait qu'un programme musical, qu'il soit écrit par un enseignant ou un étudiant, doit d'une part être communiqué, doit déboucher aussi sur une musique audible. C'est pourquoi je donne tant d'importance à l'organisation d'auditions et de concerts, et à l'existence de recueils écrits de programmes commentés, quel qu'en soit le niveau. C'est en ce sens que nous avons créé un périodique à parution aléatoire, voué à l'informatique artistique : ARTINFO/MUSINFO.

*

* *

Second type de difficulté : celles qui sont liées à l'implantation universitaire.

Contentons nous de citer au passage une question qui se pose à chacun : va-t-on payer une activité aussi onéreuse par rapport à ses maigres fruits? Et aussi : qui va payer? Car une administration conçoit

mal qu'une matière réputée littéraire ou artistique réclame un budget qui l'aligne plutôt sur les disciplines scientifiques.

Rappelons simplement qu'une université devrait pouvoir tirer une notoriété des concerts donnés sous son nom, comme elle fait avec les articles scientifiques qui y sont réalisés.

Un aspect très positif de l'université reste néanmoins qu'elle permet à des compositeurs de vivre en se livrant à leur recherche personnelle. Mais la pratique enseignante n'est pas sans contraintes : il faut satisfaire dans le même temps des étudiants en musique débutant en informatique, et des étudiants en informatique non-musiciens (et non-susceptibles de recevoir un apprentissage musical de base, que l'université n'est d'ailleurs pas là pour assurer).

J'essaye de contourner ce premier écueil en proposant autant que possible des voies d'accès diversifiées à l'informatique musicale. A l'étudiant en musique, souvent déjà muni d'un projet musical, on conseillera très vite de mettre en chantier un programme personnel, et d'apprendre autant que possible la programmation "sur le tas", au fur et à mesure de ses besoins, *mutatis mutandis*. On insistera aussi sur l'analyse de programmes existants, propre à montrer quel types d'algorithmes ont été mis en jeu par des compositeurs devant telle ou telle nécessité compositionnelle. L'étudiant en musique se constituera ainsi son propre stock d'algorithmes, en partant du musical. Avec l'étudiant en informatique, l'accent sera mis sur l'écoute de bandes magnétiques, et sur la manipulation concrète du son : étant déjà familiarisé avec les organes de calcul, il lui faut découvrir les périphériques musicaux et apprendre à en *jouer*. L'approche ludique permet dans bien des cas de contourner des sentiments d'impuissance.

Encore une chose : la vocation primordiale de l'université est de former d'une part des enseignants en musique pour les écoles secondaires, d'autre part des informaticiens, en tout cas pas des compositeurs. Ce type de contradiction, peu gênant avec la majorité des étudiants, est susceptible de le devenir chez ceux des musiciens pour qui la programmation musicale tourne à la monomanie. On leur conseille généralement de passer deux licences, en musique et en informatique, qui leur assureront une subsistance de programmeur tout en leur permettant de se livrer impunément à leur obsession favorite!

★
★ ★

On finira sur un troisième ordre de difficultés propres à notre pratique : celles qui concernent directement la pédagogie. J'y distinguerai trois volets : ce que doit affronter l'étudiant en musique, ce que doit surmonter l'étudiant en informatique, pour conclure avec les tracas personnels de l'enseignant.

Les étudiants en musique se montrent en général très rapides dans l'apprentissage de la programmation : ils sont en effet habitués à la manipulation symbolique, familiers des processus temporels, munis d'un projet propre. Mais ils souffrent souvent d'un grave défaut : ils veulent aller trop vite, et s'exposent de ce fait à ce qu'on pourrait appeler "la dépression du deuxième mois". Munis de références musicales illustres

et de projets somptueux, ils sont déçus par l'extrême pauvreté musicale qui résulte d'une programmation encore balbutiante.

La thérapeutique préventive de ce genre d'accès dépressifs réside à mon avis dans la confection de système pédagogiques spécialisés, qui permettent de manipuler un matériau musical intéressant malgré des connaissances encore rudimentaires. C'est en ce sens qu'on été conçus à Vincennes des systèmes (encore trop peu nombreux) tels que KRWTH (J. Chailloux) ou BISEQUENCEUR (P. Greussay).

Autre danger : croire que leur savoir musical implicite, qui est considérable, va être pris en compte par la machine. Pour faire découvrir que l'ordinateur ne sait que ce qu'on lui fait connaître, j'insiste toujours avec le débutant sur la formulation en langue naturelle son propre savoir sur un point particulier, avant d'en organiser une représentation informatique.

Une autre difficulté pour certains, qui s'incarne par la confection de programmes démunis de données : arriver à concevoir qu'un programme définit un processus général, renouvelable à chaque exécution. L'idée de paramétrage d'une pièce de musique est en effet difficile pour certains musiciens.

On se trouve enfin, assez rarement il est vrai, devant un blocage de certains musiciens face à la programmation, ou de résistance à la programmation ("Je comprends, mais je ne peux pas"). Un tel phénomène me paraît souvent lié à l'impossibilité de formaliser, donc de réduire un problème musical. Comment préserver en effet ses utopies musicales personnelles si on commence à les couper en tranches, et à ramener à des automates finis ce qui n'y est pas réductible?

Ce genre de blocage me semble susceptible d'être évité si l'étudiant est mis en contact dès le premier jour avec la machine, et fait tourner son premier programme à l'issue du premier cours. Il faut s'amuser avec la machine sur un mode dédramatisé, sans laisser s'installer un décalage entre savoir théorique et pratique de la programmation, qui n'irait qu'en s'aggravant. La confection de recueils d'initiation, où tout exercice, le plus simple soit-il, est à pretexte musical s'est montrée dans ce domaine très utile.

Quant au risque d'auto-contemplation par auto-simulation évoqué plus haut, il ne résiste habituellement pas à l'existence de périphériques sonores ou visuels, qui donnent une limite dans le réel, et amènent à être vu ou entendu par un public.

L'étudiant en informatique est confronté à des difficultés d'un tout autre ordre : hormis le cas de celui qui est là pour être crédité d'une assiduité, et dont il convient de vaincre la passivité, on pourrait les résumer par ces phrases cent fois entendues :

- "Je ne suis pas artiste"
- "Je n'y connais rien en musique"
- "Je n'ai pas d'idées musicales à programmer".

L'enseignant doit alors s'efforcer de développer un certain nombre de points de vue. C'est quelquefois fort long et demande dans certains cas de faire un gros effort de persuasion :

- il faut arriver à prouver à chacun qu'il porte en lui de la musique, et que son savoir musical transmis oralement est important
- il faut vaincre les problèmes liés à la notation : montrer que la notation traditionnelle n'est ni exhaustive ni absolue, que la notation musicale ne rend compte que d'une partie de

la musique qui en sera tirée (toute notation n'est que partielle), qu'une notation rudimentaire est susceptible d'engendrer des effets audibles importants, qu'on peut inventer enfin une notation musicale pour ses propres besoins. Tout ceci se prouvera sur des instruments acoustiques, avec des exemples audibles.

Ces difficultés sont encore majorées chez des étudiants du tiers-monde, nombreux à Vincennes, que l'on doit aider à se débarasser de l'idée (post-coloniale) du caractère universel et absolu de la musique de tradition occidentale, et de sa notation.

J'ai été souvent aidé en conseillant aux étudiants de travailler en paire (un étudiant en musique, un en informatique) et en préconisant le détournement des algorithmes : on peut souvent trouver des applications musicales à des programmes non musicaux déjà existants et bien contrôlés par l'étudiant en informatique.

Je finirai ce trop long exposé en évoquant quelques unes des difficultés que j'ai rencontrées en tant qu'enseignant.

Il est d'abord difficile d'être toujours intéressant : on n'est intéressant que si l'on parle avec passion, et l'on est passionné que par ce qu'on est en train de faire. Mais l'informatique musicale est récente et va très vite. On ne saurait y fournir à l'étudiant des "bases", sorte de savoir collectif commun à tous ses praticiens. On ne peut donc enseigner qu'un savoir relié aux recherches courantes, en sachant qu'on ne peut les totaliser et que ses propres spéculations sortent parfois du champ d'intérêt des élèves. C'est dire l'importance de dispositifs ouverts sur la recherche internationale : bibliothèque, abonnements à des revues, prêt d'articles, copies de bandes. Car même si l'enseignant est en retard sur le progrès, il faut que l'étudiant puisse avoir accès à ce qui se passe ailleurs, et qui peut répondre à ses préoccupations personnelles.

Encore quelques indications, en vrac : quel langage de programmation? Toujours deux simultanément à mon sens : un langage évolué (à Vincennes LISP ou FORTRAN) et un assembleur, en s'appuyant sur un recueil d'exercices multilingues : chaque exercice est écrit dans plusieurs langages. Quel contenu donner au cours? Etre aussi varié que possible : programmation théorique et pratique, analyse de programmes et d'oeuvres, écoute de musique, panorama international. Sans oublier l'abord historique que les étudiants apprécient toujours jusque dans l'anecdote : quelle est la place de la *computer music* dans la musique contemporaine? Quelle est la place de tel compositeur dans la *computer music*?

Je soumets enfin au lecteur deux questions que je tiens à laisser ouvertes, et qui touchent à la spécificité ou non de la musique à l'ordinateur :

- l'analyse d'une oeuvre composée avec ordinateur peut-elle se réduire à l'analyse du programme qui l'a engendrée?
- doit-on enseigner la composition musicale avec ordinateur comme une discipline indépendante ? Autrement dit : les méthodes traditionnelles de composition musicale sont-elles une base nécessaire, ou au contraire un fardeau inutile pour le compositeur qui travaille sur ordinateur?

Arrivé à la coda, je m'aperçois qu'il est nécessaire de modifier mon incipit. J'y remerciais un ami, mais il convient avant tout de remercier les étudiants de Vincennes. Car lui et moi leur sommes redevables, directement ou indirectement, des quelques idées exprimées dans cette communication.

AVERTISSEMENT

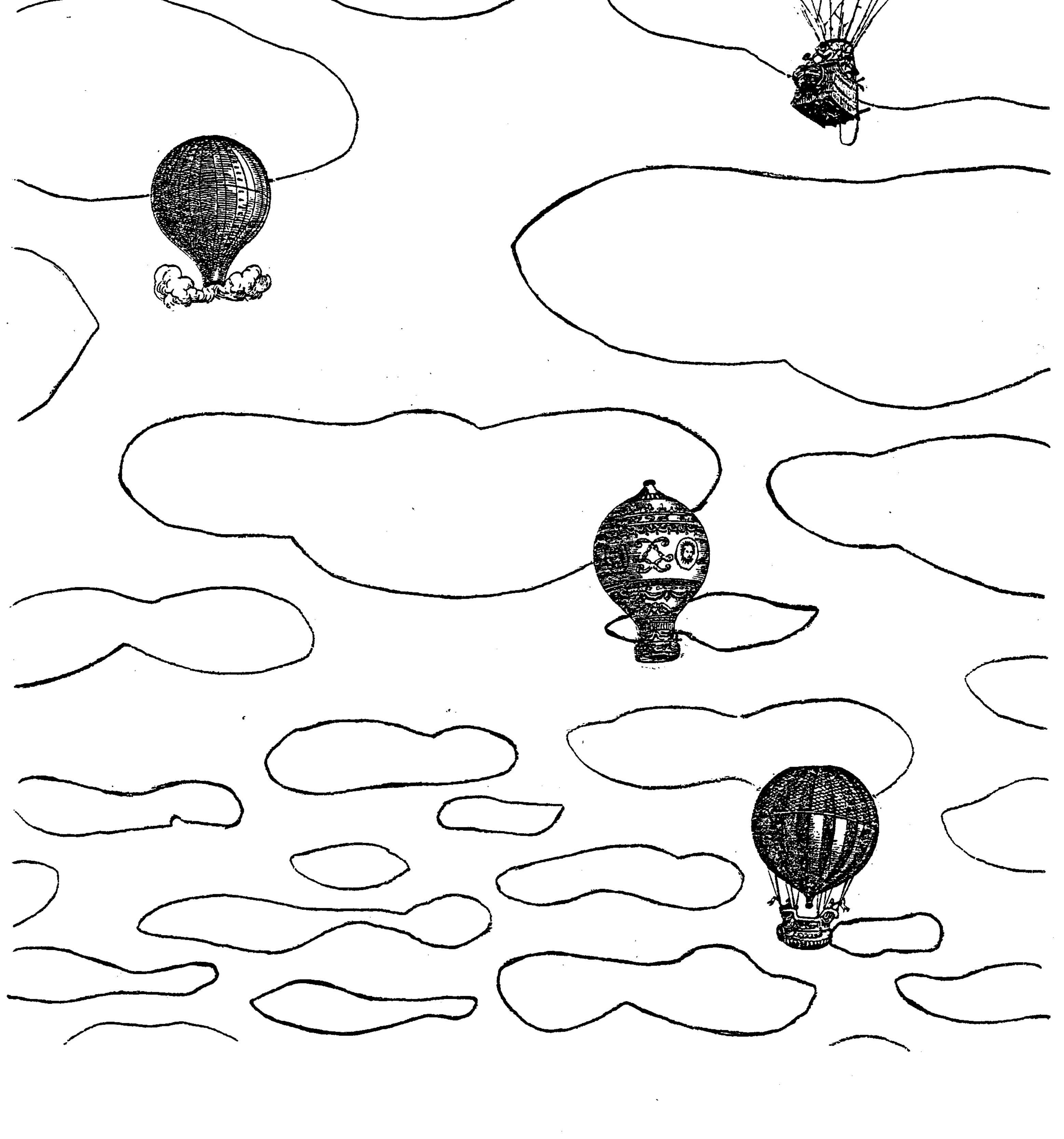
Le présent bulletin répond à une visée toute didactique : livrer sous une forme accessible aux nouveaux venus dans les groupes de travail courants de l'information technique et bibliographique en rapport avec leurs disciplines ; des programmes commentés de tous niveaux permettant un accès rapide à des techniques de programmation appropriées, ainsi qu'à une implémentation aisée.

On s'est efforcé, dans la plus large mesure possible, de ne pas établir de clivage trop marqué entre les disciplines intéressées (musique, arts plastiques, poésie, logique, architecture, cinéma, informatique), mais tout au contraire de les unifier, ne serait-ce que par des techniques de programmation communes.

L'aspect pédagogique d'ARTINFØ/MUSINFØ reflète une préoccupation constante du groupe, à savoir ne pas se satisfaire en dernier ressort de méthodes de programmation trop élémentaires.

ARTINFØ/MUSINFØ est imprimé au Département d'Informatique de l'Université de Paris VIII-Vincennes. Pour tous renseignements et composition des livraisons à venir, s'adresser à Jacques ARVEILLER, Département d'Informatique, Université de Paris VIII, 2 rue de la Liberté, 93526 SAINT DENIS CEDEX 02. Pour tout envoi, s'adresser à Patrick GREUSSAY, même adresse.

*



ARTINFO / MUSINFO #30

UNIVERSITÉ PARIS VIII

GROUPE ART & INFORMATIQUE

1980